The complexity of probabilistic inference in multi-dimensional Bayesian classifiers

Abstract

Multi-dimensional Bayesian networks (MBCs) have been recently shown to perform efficient classifications. In this study, we evaluate the computational complexity of exact inference, MAP (maximum a posterior) and MPE (most probable explanation) in MBCs. Even when MBCs have simple graphical structures under strong constraints, we find that computing exact inference is *NP-Complete*, while computing *MAP* and *MPE* is *NP-hard*.

Key words: Multi-dimensional Bayesian classifier, graphical model, exact inference, *MAP*; *MPE*

1. Introduction

Multi-dimensional Bayesian network classifiers *(MBC)* have been proven to solve the prediction problems efficiently [1, 13, 14]. The merit of MBC is that they not only recover the relationships between feature variables, but also between class variables. MBC improves the prediction performance by taking into account the conditional independencies between class variables [1]. While many learning algorithms have been proposed to train *MBC* [1, 13, 14], the fundamental difficulty to use large multi-dimensional Bayesian classifiers relies on the probabilistic inference. Nevertheless, the complexity of probabilistic inference in *MBC* remains unclear. In this study, we investigated the computational complexity of probabilistic inference in the multi-dimensional Bayesian classifiers.

The paper is organized as follows. Section two defines multi-dimensional Bayesian network classifier and its probabilistic inference problem. Section three shows that exact inference in MBC is *NP-Complete*. Section four evaluates the complexity of gray-code-based inference. Section five discusses the related work. Section six concludes the significance of our study.

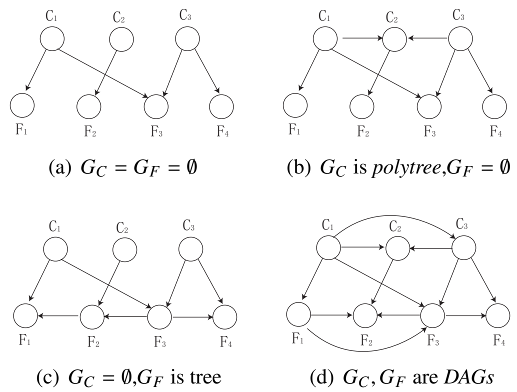
2. Definition of multi-dimensional Bayesian classifier and its probabilistic inference

2.1. Multi-dimensional Bayesian classifier

Bayesian classifiers are probabilistic models used to solve classification problems with broad applications. Following our previous study in [1], *MBC* is defined as $\mathcal{B} = (\mathcal{G}, \Theta)$, where $\mathcal{G}$ represents the graphical structure consisting of class graph G_C, feature graph G_F and bridge graph G_{CF}. Meanwhile, Θ denotes the conditional probability encoded in G. Formally, *MBC* is defined as:

$$\mathcal{B} = (\mathcal{G}, \Theta)$$

Where Θ is the set of conditional probabilities $P(X|Pa(X))$, encoded in the graphical structure $\mathcal{G}$. Specifically, the graphical structure is defined as: $\mathcal{G} = (V_C \cup V_F, E_C \cup E_{CF} \cup E_F) = G_C \cup G_{CF} \cup G_F$, where $V_C = \{C_1, C_2, ..., C_n\}$ is the set of class variables and $V_F = \{F_1, F_2, ..., F_m\}$ is the set of feature variables. Moreover, the class graph G_C is denoted by $G_C = (V_C, E_C), E_C = \{C_i \rightarrow C_j | C_i, C_j \in V_C\}$. The symbol $C_i \rightarrow C_j$ indicates that there is a direct arc from C_i to C_j but not from C_j to C_i. The feature graph G_F is denoted by $G_F = (V_F, E_F)$, where $E_F = \{F_i \rightarrow F_j | F_i, F_j \in V_F\}$. The bridge graph G_{CF} is denoted by $G_{CF} = (V_C \cup V_F, E_{CF})$, where $E_{CF} = \{C_i \rightarrow F_j | \sum_{i=1}^{n} CE(C_i, F_j) \geq 1, \sum_{j=1}^{m} CE(C_i, F_j) \geq 1, C_i \in V_C, F_j \in V_F\}$. Note that $CE(C_i, F_j) = 1$ if $C_i \rightarrow F_j$; otherwise, $CE(C_i, F_j) = 0$.



(a) $G_C = G_F = \emptyset$ (b) G_C is *polytree*,$G_F = \emptyset$

(c) $G_C = \emptyset$,G_F is tree (d) G_C, G_F are *DAGs*

Figure 1: Examples of *MBCs*

Figure1 illustrates several examples of *MBC*. In these examples, C_1, C_2 and C_3 are class (target) variables, F_1, F_2, F_3 and $F_4\}$ are feature (evidence) variables. Moreover, the class graph G_C and the feature graph G_F are formed as either trees, *polytrees* or *DAGs*. While G_C, G_F are not restricted to the above structures, they can be any structure that does not contain a cycle. Additionally, it is required that at least one of the class variables has an edge to feature variables. Overall, *MBC* is designed to solve classification problems where multi-class variables exist.

2.2. Probabilistic inference in MBC

Before we introduce probabilistic inference in *MBC*, let's review some notations in general Bayesian networks. Given a general Bayesian network, $\mathcal{B} = (\mathcal{G}, \Theta)$, where $\mathcal{G}$ represents a directed acyclic graph. Each node in $\mathcal{G}$ indicates a variable. In our study, we only consider that all variables are discrete. *MAP* (maximum a posterior) and *MPE* (most probable explanation) are two known inference problems in Bayesian networks. Suppose V_T and V_E denote the sets of target and evidence variables, respectively. *MAP* and *MPE* are defined as:

$$MAP : V_T^* = \arg\max_{V_T} P(V_T|V_E)$$
$$MPE : V_T^* = \arg\max_{V_T} P(V_T, V_E)$$

Similar to the definitions in the general Bayesian networks, we define the exact inference(*EI*), *MAP* and *MPE* in multi-dimensional Bayesian classifiers. Suppose $\mathcal{B} = (\mathcal{G}, \Theta)$ denotes a *MBC* and the assignment of input feature variables $(f_1, ..., f_m)$, we have:

$$EI : \quad P(V_C|V_F = (f_1, ..., f_m))$$
$$MAP : V_C^* = \arg\max_{V_C} P(V_C|V_F = (f_1, ..., f_m))$$
$$MPE : V_C^* = \arg\max_{V_C} P(V_C, V_F = (f_1, ..., f_m))$$

Given an instance of evidence for feature variables, *MAP* endeavors to search a configuration of all class variables which achieves the maximum posterior probability of $P(V_C|V_F)$. Similarly, *MPE* searches a configuration to maximize the joint probability $P(V_C, V_F)$ instead of the conditional probability $P(V_C|V_F)$.

3. Complexity of exact inference, *MAP* and *MPE* in multi-dimensional Bayesian classifier

It has been reported that the complexity for exact inference, *MAP* and *MPE* in general Bayesian network is *NP-hard* [2, 5, 7, 8]. Due to the fact that *MBC* is a special case of general Bayesian networks, the computational complexity of *MAP* and *MPE* in *MBC* is supposed to be lower than that of general Bayesian networks. However, we will prove that the computation complexity of both *MAP* and *MPE* in *MBC* is still *NP-hard* even in the context of strong constraints on the *MBC* parameters. Before we formally prove their complexity, we would like to recall several NP problems, where a polynomial time algorithm does not exist to solve them.

3.1. Definition of several NP problems

One-In-Three 3SAT, known as *1-in-3 SAT*, is one of the known *NP-Complete(NPC)* problems [16, 17]. Instance of *One-In-Three 3SAT* is defined as a collection of clauses formed by exactly three literals. The decision problem of *One-In-Three 3SAT* is to answer whether each clause has exactly one true literal given an instance. *One-In-Three 3SAT* is also proved to be *NPC* even when no clause contains a negative literal ([17],[LO4],p259). Here, we want to prove the *NPC* of a variantion of *One-In-Three 3SAT*, termed as *One-In-Three 3SAT(3)*, in which every variable only occurs in at most three clauses. We first define our problems and prove them based on the polynomial transformation.

Problem: *3SAT (3)* [18][p183]
Instance: Suppose U is a set of variables, C is a collection of clauses and every clause c in C satisfies $|c| = 3$. For every assignment, at most 3 clauses contain either u or $\overline{u}$.
Question: Is there a truth assignment for the clauses in C?

Problem: *One-In-Three 3SAT(3)*
Instance: Suppose U is a set of variables, C is a collection of clauses and every clause c in C satisfies $|c| = 3$. For every assignment, at most 3 clauses contain either u or $\overline{u}$.
Question: Is there a truth assignment to guarantee that every clause in C has exactly one true literal?

Problem: *One-In-Three 3SAT(4): 1P3SAT(4)*
Instance: Suppose U is a set of variables, C is a collection of clauses and every clause c in C satisfies $|c| = 3$ and no clause contains any negative literal. For every assignment, at most 4 clauses contain u.
Question: Is there a truth assignment to guarantee that every clause in C has exactly one true literal?

Problem: *One-In-Three 3SAT Variant(4): 1P3SATV(4)*
Instance: Suppose U is a set of variables, C is a collection of clauses and every clause c in C satisfies $|c| = 3$ and no clause contains any negative literal. Every literal in clause c is in the form of x_i^j where $x_i \in U$ and the superscript j indicates the assignment of the j^{th} variable at x_i. For every assignment, there are at most 4 clauses contain x_i.
Question: Is there a truth assignment to guarantee that each clause in C has exactly one true literal and all $x_i^j (j = 1, 2, ..., L_i)$ are assigned with the same boolean value?

Theorem 1. *One-In-Three 3SAT(3)* is *NP-Complete*.
Proof. It is obvious a *NP* problem. To prove that it is *NP-Complete*, we construct a polynomial reduction from *3SAT(3)* to *One-In-Three 3SAT(3)*. *3SAT(3)* is *NP-Complete* based on the proposition 9.3 in [18][p183]. Given any instance $S = (U, C)$ from *3SAT(3)*, the following procedure [16] guarantees that any clause $(X \vee Y \vee Z)$ in S can be transformed into a clause $G(X, Y, Z)$ of *One-In-Three 3SAT(3)*.

$$(X \vee Y \vee Z) \Rightarrow G(X, Y, Z) = (X \vee a \vee b) \wedge (Y \vee b \vee d) \wedge$$
$$(a \vee b \vee e) \wedge (c \vee d \vee f) \wedge (Z \vee c \vee False)$$

We have converted an instance S from *3SAT(3)* into S' in *One-In-Three 3SAT (3)* through a polynomial reduction. Five new variables (a, b, c, d, e, f) are added for each clause $c \in C$ so that S is transformed into a new instance S' of *One-In-Three 3SAT(3)*. Note that (X, Y, Z) has a truth assignment in *3SAT(3)* if and only if $G(X, Y, Z)$ has a truth assignment in *One-In-Three 3SAT(3)*. We conclude that *One-In-Three 3SAT(3)* is also *NP-Complete* since *3SAT(3)* is *NP-Complete*. □

Theorem 2.*One-In-Three 3SAT(4)* is *NP-Complete*.

Proof. It is obvious a *NP* problem. We now derive a polynomial reduction to transform any instance from *One-In-Three 3SAT(3)* into an instance of *One-In-Three 3SAT(4)*. Given one clause $c = (x, y, z)$ in an instance S from *One-In-Three 3SAT(3)*, it contains at most one negative literal and we assume that the first literal x is negative:

$$U = \{x_1, x_2, ..., x_n\}, C = \{C_1, C_2, ..., C_m\}, |C_i| = 3.$$

$$H^C(x_i) = \sum_{j=1}^{n} H^{C_j}(x_i) \leq 3, i = 1, ..., n.$$

$$Where\ H_{C_j}(x_i) = \begin{cases} 1 & x_i\ or\ \overline{x_i} \in C_j \\ 0 & Otherwise \end{cases}, j = 1, ..., m.$$

For every x_i in U, if x_i occurs as negative literals for three times, we replace $\overline{x_i}$ with a new variable y_i. If $\overline{x_i}$ occurs more than once, we transform every clause with the negative literal $\overline{x_i}$ as follows:

$$(\overline{x_i} \vee a \vee b) \Rightarrow (y_i \vee a \vee b) \wedge (y_i \vee x_i \vee False)$$

Note that the transformed S' is an instance of *Strict-One-In-Three 3SAT(4)* and obeys the rule that every variable in S' occurs at most four times. Moreover, $(\overline{x_i} \vee a \vee b)$ is true if and only if $(y_i \vee a \vee b) \wedge (y_i \vee x_i \vee False)$ is true. Hence S has a truth assignment if and only if S' has a truth assignment. *One-In-Three 3SAT(4)* is *NP-Complete* since *One-In-Three 3SAT(3)* is *NP-Complete*. □

Theorem 3. *One-In-Three 3SAT Variant(4)* is *NP-Complete*.

Proof. Membership in *NP* is immediate. For any instance S of *1P3SAT(4)*, it can be transformed into an instance S' of *1P3SATV(4)* by replacing x_i with x_i^j, where j is the j^{th} variable in x_i. Based on the transformation proposed in [9], we have

$$\begin{aligned} S &: \quad (x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee x_5) \\ S' &: \quad (x_1^1 \vee x_3^1 \vee x_4^1) \wedge (x_1^2 \vee x_2^1 \vee x_4^2) \wedge (x_2^2 \vee x_3^2 \vee x_5^1) \end{aligned}$$

Above transformation can be done within polynomial time. Obviously, S has a truth assignment if and only if S' has a truth assignment. *1P3SATV(4)* is *NP-Complete* since *1P3SAT(4)* is *NP-Complete*. □

3.2. Exact inference in multi-dimensional Bayesian classifiers

Given a *MBC* $\mathcal{B} = (\mathcal{G}, \Theta)$, its exact inference is defined as:

$$P(C_1, C_2, ...C_n | F_1, ..., F_m) = \frac{P(C_1, C_2, ...C_n, F_1, ..., F_m)}{P(F_1, ..., F_m)}$$

Moreover, the joint probability in *MBC* can be decomposed by conditional probabilities.

$$P(V) = \prod_{X \in V} P(X | Pa(X))$$

We first focus on the simplest exact inference in *MBC* where none of the class variable in *MBC* has any parent node.

Theorem 4. The exact inference is *NP-Complete* even when the parameters in *MBC* $\mathcal{B} = (\mathcal{G}, \Theta)$ satisfy:

$$(a) \quad G_C = G_F = \emptyset, \sum_{i=1}^{n} CE(C_i, F_j) \leq 6.$$

$$(b) \quad G_C\ and\ G_F\ are\ trees,\ \sum_{i=1}^{n} CE(C_i, F_j) = 1.$$

Where $CE(C_i, F_j) = 1$ if $C_i \rightarrow F_j \in E$; otherwise, $CE(C_i, F_j) = 0, i = 1, ..., n, j = 1, ..., m.$

Proof. We use *1P3SAT(4)* and *1P3SATV(4)* to prove (a) and (b), respectively. Suppose $S = (U, C^*)$ be an arbitrary instance of problem *1P3SAT(4)*, we present a polynomial transformation to reconstruct a *MBC* from S.

Construction 1: *1P3SAT(4)* to *MBC*.
Input: An arbitrary instance $S = (U, C^*)$ of *1P3SAT(4)*,
$U = (X_1, X_2, ..., X_m), C^* = (C_1^*, C_2^*, ..., C_n^*)$.
Output: a reconstructed *MBC*: $\mathcal{B} = (\mathcal{G}, \Theta)$.
Step 1: For every clause $C_i^* \in C^*$, we construct a class variable C_i in $\mathcal{G}$. Similarly, for every variable $X_i \in U$, one feature variable F_i is created in $\mathcal{G}$. Given a clause $C_i^* = (X_{i1}, X_{i2}, X_{i3})$, a bridge graph is reconstructed by adding the arcs $C_i \rightarrow F_{i1}, C_i \rightarrow F_{i2}, C_i \rightarrow F_{i3}$. By doing so, F_{i1}, F_{i2}, F_{i3} are mapped to X_{i1}, X_{i2}, X_{i3}. All C_i and F_i are binary variables.
Step 2: To map every clause to *MBC*, we add n extra nodes $\{J_1, ..., J_n\}$ as feature variables. We also create additional arcs to link feature variables with class variables: $C_1 \rightarrow J_1, C_{i-1} \rightarrow J_i, C_i \rightarrow J_i$ where $i = 2, ..., n$.
Step 3: The probability distribution Θ of *MBC* is defined as:
Class variables :

$$\begin{aligned} P_i : & P(C_i = 0 | J_i = 0) = 1 \\ & P(C_i = 1 | J_i = 1, F_{i1} \vee F_{i2} \vee F_{i3} = 1) = 1 \end{aligned}$$

Feature variables:

$$\begin{aligned} P_i^{\dagger} : & P(F_i = 1) = P(F_i = 1) = \frac{1}{2}, i = 1, 2, ..., m. \\ P_i^{\ddagger} : & P(J_1 = 1) = 1. \\ & P(J_i = 1 | C_{i-1} = 1) = 1, i = 2, ..., n. \\ & P(J_i = 0 | C_{i-1} = 0) = 1, i = 2, ..., n. \end{aligned}$$

Given any instantiation of *1P3SAT(4)*, the parameters in the reconstructed $\mathcal{B} = (\mathcal{G}, \Theta)$ are:

$$\begin{aligned} & \mathcal{G} = (V, E), V = \{C_1, ..., C_n, F_1, ..., F_m, J_1, ..., J_n\} \\ & E = (\bigcup_{i=1}^{n} \{C_i \rightarrow F_{i1}, C_i \rightarrow F_{i2}, C_i \rightarrow F_{i3}\}) \cup \\ & (C_1 \rightarrow J_1 \bigcup_{i=2}^{n} \{C_{i-1} \rightarrow J_i, C_i \rightarrow J_i\}) \\ & \Theta = (\bigcup_{i=1}^{n} P_i) \cup (\bigcup_{i=1}^{m} P_i^{\dagger}) \cup (\bigcup_{i=1}^{n} P_i^{\ddagger}) \end{aligned}$$

Above transformation is polynomial. We provide a simple example to explain above procedure. Give S =

$(U, C^*) \in 1P3SAT(4), U = \{X_1, X_2, X_3, X_4, X_5\}, C^* = \{(X_1, X_2, X_3), (X_1, X_3, X_4), (X_3, X_4, X_5)\}$, the transformed *MBC* is shown in Figure 2.

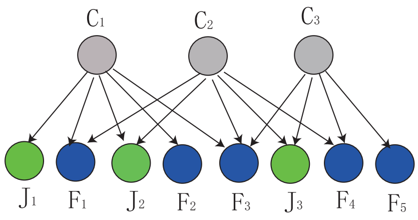


Figure 2: The reconstructed *MBC* by construction 1.

Moreover, using the transformation above leads to the following result: $P(C_n = 1) = \frac{\sharp C_n}{2^{3n}}$, where $\sharp C_n$ is the number of true assignments that satisfy the instance S, n is the number of clauses. The complete proof is included in Theorem 7 (see appendix). Based on the above formula, the instance S has a truth assignment if and only if $P(C_n = T) > 0$. Moreover, if S has a truth assignment, $\sharp C_n \geq 1$ so that $P(C_n = T) > 0$. Since a variable appears at most four times in *1P3SAT(4)* and every class variable connects with two additional feature variables, therefore $\sum_{i=1}^{n} CE(C_i, F_j) \leq 6, j = 1, ..., m.$, We thus conclude that exact inference in *MBC* under the condition of (a) is *NP-Complete*.

(b) We use *1P3SATV(4)* to prove that the exact inference of *MBC* is *NP-Complete* even if both G_C, G_F are tree models.

Construction 2: *1P3SATV(4)* to *MBC*
Input: An instance $S = (U, C^*)$ from *1P3SATV(4)*,
$U = (X_1, X_2, ..., X_m)$, $C^* = (C_1^*, C_2^*, ..., C_n^*)$.
Output: a reconstructed *MBC*: $\mathcal{B} = (\mathcal{G}, \Theta)$.
Step 1: For every clause C_i^*, we create a class variable C_i and every variable X_i in S is mapped to a feature variable F_i in *MBC*.
Step 2: The class graph G_C is reconstructed by adding the class variables into a chain $C_1 \rightarrow C_2, ..., \rightarrow C_n$. Feature graph G_F is created by adding arcs between variables that share the same subscript in C^*. Bridge graph G_{CF} is reconstructed by adding arcs $C_i \rightarrow F_{i1}, C_i \rightarrow F_{i2}, C_i \rightarrow F_{i3}$. Note that F_{i1}, F_{i2}, F_{i3} are mapped to X_{i1}, X_{i2}, X_{i3} where $C_i^* = (X_{i1}, X_{i2}, X_{i3}), i = 1, ..., n$. Both F_i and C_i are binary variables in *MBC*.
Step 3: The probability distribution Θ of *MBC* is defined as:
Class variables:

$$
\begin{aligned}
P_i^{\dagger} : \; & P(C_i = 0 | C_{i-1} = 0) = 1 \\
& P(C_1 = 1 | F_{11} \vee F_{12} \vee F_{13} = 1) = 1 \\
& P(C_i = 1 | C_{i-1} = 1, F_{i1} \vee F_{i2} \vee F_{i3} = 1) = 1
\end{aligned}
$$

Feature variables :

$$
\begin{aligned}
P_i^{\ddagger} : \; & P(F_i = 1) = P(F_i = 1) = \frac{1}{2} \\
& P(F_i = 1 | F_i^* = 1) = 1, P(F_i = 0 | F_i^* = 0) = 1
\end{aligned}
$$

Given any instantiation S of *1P3SATV(4)*, $S = (U, C)$, $U = \{X_1, X_2, ..., X_m\}$, $C^* = \{C_1^*, C_2^*, ..., C_n^*\}$, the parameters in the $\mathcal{B} = (\mathcal{G}, \Theta)$ are :

$$
\begin{aligned}
& \mathcal{G} = (V, E). \\
& V = \{C_1, ..., C_n, F_1, ..., F_m\}, E = E_C \cup E_{CF} \cup E_F. \\
& E_C = \bigcup_{i=1}^{n-1} (C_i \rightarrow C_{i+1}), E_F = \bigcup_{i=1}^{L} (\bigcup_{j=1}^{L(i-1)} F^{i(j)} \rightarrow F^{i(j+1)}). \\
& E_{CF} = \bigcup_{i=1}^{n} (C_i \rightarrow F_{i1}, C_i \rightarrow F_{i2}, C_i \rightarrow F_{i3}). \\
& \Theta = (\bigcup_{i=1}^{n} P_i^{\dagger}) \cup (\bigcup_{i=1}^{m} P_i^{\ddagger}).
\end{aligned}
$$

Where $F^{i(1)}, ..., F^{i(L_i)}$ in $\bigcup_{i=1}^{L} (\bigcup_{j=1}^{L_i-1} F^{i(j)} \rightarrow F^{i(j+1)})$ are feature variables that take the same subscript i in the chain of $F^{i(1)} \rightarrow \cdots \rightarrow F^{i(L_i)}$. L is the number of different chains(trees) in feature graph G_F. L_i is the length of the i^{th} chain in the G_F. One example is: $S = (U, C^*) \in$ *1P3SATV(4)*.

$$
\begin{aligned}
& U = \{x_1^1, x_1^2, x_1^3, x_2^1, x_2^2, x_3^1, x_3^2, x_4^1, x_4^2\} \\
& C^* = \{\{x_1^1, x_3^1, x_4^1\}, \{x_1^2, x_2^1, x_4^2\}, \{x_1^3, x_2^2, x_3^2\}\}
\end{aligned}
$$

The reconstructed structure of *MBC* is shown in Figure 3.

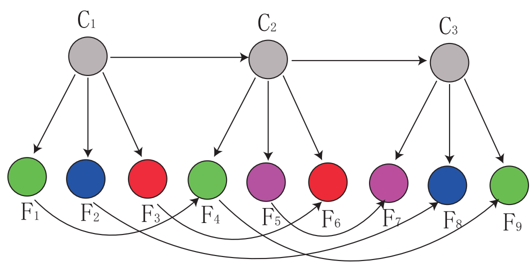


Figure 3: The reconstructed *MBC* by construction 2.

Above transformation can be accomplished within polynomial time $O(Max(n, m))$. Based on the reconstructed structure, we have: $P(C_n = 1) = \frac{\sharp C_n}{2^L}$, where $\sharp C_n$ is the number of true assignments that satisfy the instance S, L is the number of chains (trees) in the feature graph G_F. The complete proof of above formula is given in Theorem 8 (see appendix). Based on above formula, instance S has a true assignment if and only if $P(C_n = T) > 0$. Since any variable in *1P3SATV(4)* occur at most four times, we have $\sum_{i=1}^{n} CE(C_i, F_j) \leq 4, j = 1, ..., m.$. We thus conclude that the exact inference in *MBC* under the condition of (b) is *NP-Complete*. □

3.3. MAP and MPE in multi-dimensional Bayesian classifier

This section proves that *MAP* and *MPE* in *MBC* is *NP-hard*.

Theorem 5. Given a *MBC* $\mathcal{B} = (\mathcal{G}, \Theta)$, *MAP* is *NP-hard* even when $\mathcal{G}$ is restricted to:

$$(a) \quad G_C = G_F = \emptyset, \sum_{i=1}^{n} CE(C_i, F_j) \leq 4.$$

$$(b) \quad G_C = \emptyset, G_F \ are \ forest, \ \sum_{i=1}^{n} CE(C_i, F_j) = 1.$$

Where $CE(C_i, F_j) = 1$ when $C_i \rightarrow F_j \in E$; otherwise $CE(C_i, F_j) = 0, i = 1, ..., n, j = 1, ..., m$.

Proof. We prove *MAP* of *MBC* is *NP-hard* by the reduction from *NPC* problem *1P3SAT(4)* and *1P3SATV(4)*. Decision problem of *MAP* claims: given an evidence of feature variables $(f_1, ..., f_m)$ and a constant α, is there an instantiation of class variable such that: $P(C = (c_1, ..., c_n)|F = (f_1, ..., f_m)) \geq \alpha$?

(a) Given an instance of *1P3SAT(4)*, the reduction can be performed by the following procedure.

Construction 3: *1P3SAT(4)* to *MBC*
Input: an arbitrary instance $S = (U, C^*)$ of *1P3SAT(4)*,
$U = (X_1, X_2, ..., X_n), C^* = (C_1^*, C_2^*, ..., C_m^*)$.
Output: $\mathcal{B} = (\mathcal{G}, \Theta)$.
Step 1: For each clause C_i^*, we construct a class variable C_i and for every variable X_i, we create one feature variable F_i. Both class and feature graphs are empty. For each clause $C_i^* = (X_{i1}, X_{i2}, X_{i3})$, the bridge graph is reconstructed by adding the arcs $C_i \rightarrow F_{i1}, C_i \rightarrow F_{i2}, C_i \rightarrow F_{i3}$. Both class variables C_i and F_i are binary variables in *MBC*.
Step 2: We define the probability distribution of *MBC* as:
Class variables :

$$P_i^{\dagger} : P(C_i = 1|F_{i1} \vee F_{i2} \vee F_{i3} = 1) = 1$$
$$P(C_i = 0|F_{i1} \vee F_{i2} \vee F_{i3} = 0) = \frac{1}{2}$$

Feature variables :

$$P_i^{\ddagger} : \ P(F_i = 1) = P(F_i = 0) = \frac{1}{2}, i = 1, ..., m$$

Given $S = (U, C^*) \in 1P3SAT(4), U = \{x_1, x_2, x_3, x_4, x_5\}, C^* = \{(x_1, x_2, x_3), (x_1, x_3, x_4), (x_3, x_4, x_5)\}$, Figure 4 shows the transformed *MBC*. Moreover, above procedure to construct *MBC*

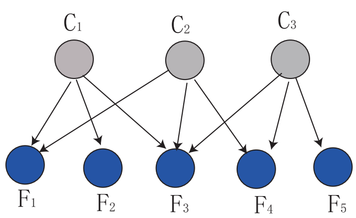


Figure 4: The reconstructed *MBC* by construction 3.

can be accomplished within polynomial time. Due to the fact that C_i and C_j are independent, we have:

$$P(C_1, ..., C_n|F_1, ..., F_m) = \prod_{i=1}^{n} P(C_i|F_1, F_2, ..., F_m)$$
$$= \prod_{i=1}^{n} P(C_i|F_{i1}, F_{i2}, F_{i3})$$

Note that if $P(C_1, ..., C_n|F_1 = f_1, ..., F_m = f_m) = 1$, all class variables from the solution set of *MAP* are assigned true; otherwise, $P(C_1, ..., C_n|F_1 = f_1, ..., F_m = f_m) < 1$. It indicates that the instance S has a true assignment if and only if $P(C_1 = 1, ..., C_n = 1|F_1 = f_1, ..., F_m = f_m) = 1$. Since every variable in the instance S occur at most four times, $\sum_{i=1}^{n} CE(C_i, F_j) \leq 4, j = 1, ..., m$. Therefore the decision problem of *MAP* is *NP-hard*.

(b) To prove the second result, we used similar reduction as part (a).
Construction 4: *3SATV(4)* to *MBC*
Input: An arbitrary instance $S = (U, C^*)$ of *3SATV(4)*,
$U = (X_1, X_2, ..., X_m), C^* = (C_1^*, C_2^*, ..., C_n^*)$.
Output: $\mathcal{B} = (\mathcal{G}, \Theta)$
Step 1: For every clause C_i^*, we create a class variable C_i. For every variable X_i in S, we create a feature variable F_i. No arc is added into the class graph and $G_C = \emptyset$. The feature graph G_F is reconstructed by adding arcs between variables that have the same subscript. Specifically, for every clause $C_i = (X_{i1}, X_{i2}, X_{i3})$, a bridge graph is constructed by adding arcs $C_i \rightarrow F_{i1}, C_i \rightarrow F_{i2}, C_i \rightarrow F_{i3}$. Both C_i and F_i are binary variables in *MBC*.
Step 2: $I(F_1, ..., F_m) = 1$ indicates that feature variables in the same chain of G_F take the same binary value. We define the probability distribution of *MBC* as:
Class variables :

$$P_i^{\dagger} : P(C_i = 1|F_{i1} \vee F_{i2} \vee F_{i3} = 1, I(F_1, ..., F_m) = 1) = 1.$$
$$P(C_i = 0|F_{i1} \vee F_{i2} \vee F_{i3} = 0, I(F_1, ..., F_m) = 1) = \frac{1}{2}$$

Feature variables :

$$P_j^{\ddagger} : P(F_j = 1) = P(F_j = 0) = \frac{1}{2}, j = 1, ..., m.$$

We give an example to show how this reduction works. Suppose an instance $S = (U, C^*)$ of *1P3SATV(4)* is defined as:

$$U = \{x_1^1, x_1^2, x_2^1, x_2^2, x_3^1, x_3^2, x_4^1, x_4^2, x_4^4, x_5^1\}$$
$$C^* = \{\{x_1^1, x_3^1, x_4^1\}, \{x_1^2, x_2^1, x_4^2\}, \{x_2^2, x_3^2, x_5^1\}\}$$

Figure5 shows the transformed structure. Moreover, constructing a *MBC* is accomplished in polynomial time. Based on the construction 4, we have:

$$P(C_1, ..., C_n|F_1, ..., F_m) = \prod_{i=1}^{n} P(C_i|F_1, F_2, ...F_m)$$

$P(C_1, ..., C_n|F_1 = f_1, ..., F_m = f_m) \geq \alpha = 1$ is true if and only if the query of *3SATV(4)* on $X_1 = f_1, ..., X_m = f_m$

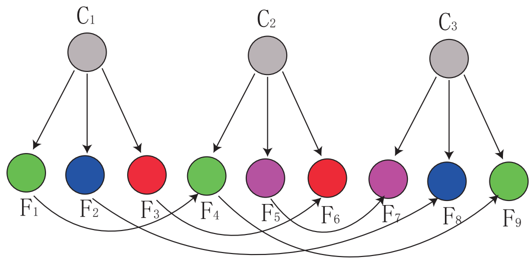


Figure 5: The reconstructed *MBC* by construction 4.

is true. Therefore, even under the constraints of $\sum_{i=1}^{n} CE(V_{C_i}, V_{F_j}) = 1, j = 1, ..., m$, *MAP* is *NP-hard*. □

Theorem 6. Given a *MBC* $\mathcal{B} = (\mathcal{G}, \Theta)$, *MPE* is *NP-hard* even when $\mathcal{G}$ satisfies:

$$(a) \quad G_C = G_F = \emptyset, \sum_{i=1}^{n} CE(C_i, F_j) \leq 4.$$

$$(b) \quad G_C = \emptyset, G_F \; are \; forest, \sum_{i=1}^{n} CE(C_i, F_j) = 1.$$

Where $CE(C_i, F_j) = 1$ if $C_i \rightarrow F_j \in E$; otherwise, $CE(C_i, F_j) = 0, j = 1, ..., m, j = 1, ..., n$.

Proof. The joint probability in *MBC* satisfies:

$$P(C_1, ..., C_n, F_1, ..., F_m) = P(C_1, ..., C_n|F_1, ..., F_m)P(F_1, ..., F_m)$$

(a) The feature graph G_F in part (a) of Theorem 5 enables us to decompose $P(F_1, F_2, ..., F_m)$ because all feature variables are independent. Moreover, $P(F_i = 1) = P(F_i = 0) = 0.5, i = 1, .., m$ leads to:

$$\begin{aligned} P(C_1, ..., C_n, F_1, ..., F_m) &= P(C_1, ..., C_n|F_1, ..., F_m) \prod_{i=1}^{m} P(F_i) \\ &= (\frac{1}{2})^m \prod_{i=1}^{n} P(C_i|F_{i1}, F_{i2}, F_{i3}) \end{aligned}$$

Let $\alpha = (\frac{1}{2})^m$ be the parameter in the decision problem of *MPE*, the rest of the analysis is similar to the part (a) in Theorem 5.

(b) Based on the analysis in the part (b) of Theorem 5 and Theorem 8 (see appendix), we have:

$$\begin{aligned} P(C_1...C_n, F_1...F_m) &= P(C_1...C_n|F_1...F_m)P(F_1...F_m) \\ &= \frac{1}{2^L} \prod_{i=1}^{n} P(C_i|F_1, F_2, ...F_m) \end{aligned}$$

Let $\alpha = 1/2^L$ and L be the number of chains in the feature graph, the rest of the proof is similar to the part (b) in Theorem 5. □

4. Related work

Inference on general Bayesian networks has been investigated in many studies. Researchers have focused on designing approximate algorithms to solve *MAP* or *MPE*. To design efficient algorithms for *MAP* is crucial in most scenario. However, solving *MAP* within polynomial time is impossible in Bayesian networks. In 1987, Cooper [2] proved that probabilistic inference in belief networks is *NP-hard* by transforming a well-known *NP-hard* problem, *3SAT*, into belief networks *(PIBNET)*. In 1993, Dagum [3] proved that approximating probabilistic inference in general Bayesian networks is also *NP-hard*. In 1998, Michael L.Littman [4] proved that the complexity of probabilistic planning is *NP-hard* in different plan types [5], and approximating *MAP* in general Bayesian network within the bound of $[l, u]$ is still impossible where $0 \leq l < 0.5 < u \leq 1$. In 2002, James [6, 7] proved that *MAP* is *NPC*, even if Bayesian networks were restricted to polytrees. Moreover, approximating *MAP* within any factor $f(n)$ is surprisingly *NP-hard*, where n is the number of nodes and *MAP* of general network is in the group of $\mathcal{NP}^{\mathcal{PP}}$. In 2003, Shimony [8] proved that *MPE* in the directed-path singly-connected Bayesian networks is still *NP-hard*. In 2005, Dan Wu [9] proved that probability inference in singly connected Bayesian networks is also *NP-hard*.

The computational complexity of *MAP* and *MPE* has been investigate in in general Bayesian networks. The reduction is usually performed by transforming *SAT* or its variant problems (e.g. *MAX SAT*[7], *E-MAJSAT*[7], *One-In-Three 3SAT*[19]) into specific Bayesian networks. Besides, hamiltonian circuit - a famous *NP-hard* problem - has also been used to prove the computational complexity of maximum a posteriori probability in *DAG* is *NP-hard*[20]. To our knowledge, a clear study of the computational complexity of *MAP* and *MPE* in multi-dimensional Bayesian classifier has not been reported.

Researchers have designed efficient algorithms to solve *MAP* problems. In 1998, Vasanth Krishna [10] presented an algorithm to perform parallel exact inference based on junction tree decomposition. The branch-and-bound searching algorithms and local searching strategies (e.g. climbing and Taboo search) were also proposed to approximate *MAP* [6, 11]. In 2007, Xiaoxun [12] introduced dynamic weighting search algorithm $\mathcal{A}^*$ to solve *MAP* using the asymmetry properties in the underlying distribution. Our recent study also proposed inference algorithms using gray-code and we showed its promising performance in many applications [1].

5. Conclusions and future work

In this study, we investigated the computational complexity of exact inference, *MAP*,*MPE* in multi-dimensional Bayesian classifiers. We used the variations of *3SAT* to prove exact inference in (*MBC*) is *NP-Complete*. More importantly, we proved that *MAP* and *MPE* in *MBC* is *NP-hard* even under strict constraints. Future work need to design efficient approximation algorithms for fast inference in *MBC*.

6. Appendix

Theorem 7. Suppose $S = (U, C^*)$ is an instance in 1P3SAT(4) where $U = (X_1, ..., X_m), C^* = (C_1^*, ..., C_n^*)$. A *MBC*, denoted as $\mathcal{B} = (\mathcal{G}, \Theta)$, is constructed using the the reduction approach in Theorem 4(a). The joint probability $P(C_n)$ in $\mathcal{G}$ satisfies:

$$P(C_n = 1) = \frac{\sharp C_n}{2^{3n}}$$

Where $\sharp C_n$ is the number of true assignments that satisfies *1P3SAT(4)* and n is the number of clauses in S.

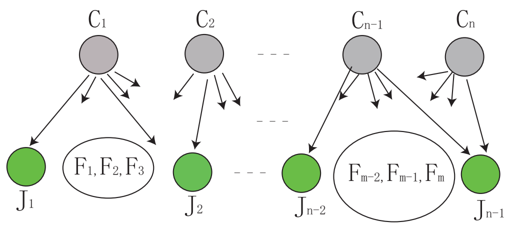


Figure 6: The schematic view of reconstructed *MBC* by construction 1.

Proof. Given the procedure in construction one of Theorem 4(a), the skeleton of the generated *MBC* is shown in Figure 6. $P(C_n = 1)$ can be measured as:

$$P(C_i = 1) = \sum_{F_{i1}, F_{i2}, F_{i3}, J_i} P(C_i = 1, F_{i1}, F_{i2}, F_{i3}, J_i) \quad (6.1)$$

$$= \sum_{F_{i1}, F_{i2}, F_{i3}} P(C_i = 1 | F_{i1}, F_{i2}, F_{i3}, J_i = 1)$$

$$P(F_{i1}, F_{i2}, F_{i3}) P(J_i = 1) \quad (6.2)$$

$$= \sum_{I_A(F_{i1}, F_{i2}, F_{i3})} P(F_{i1}, F_{i2}, F_{i3}) P(J_i = 1) \quad (6.3)$$

$$= \sum_{I_A(F_{i1}, F_{i2}, F_{i3})} P(F_{i1}, F_{i2}, F_{i3}) P(C_{i-1} = 1) \quad (6.4)$$

$$= \frac{1}{2^3} \sum_{I_A(F_{i1}, F_{i2}, F_{i3})} P(C_{i-1} = 1) \quad (6.5)$$

The $I_A(F_{i1}, F_{i2}, F_{i3})$ indicates that F_{i1}, F_{i2}, F_{i3} satisfies the condition that only one of F_{i1}, F_{i2}, F_{i3} is assigned true, the other two are false. Equation (6.3) is true because F_{i1}, F_{i2}, F_{i3} is independent with J_i and $P(F_{i1}, F_{i2}, F_{i3}, J_i = 1) = P(F_{i1}, F_{i2}, F_{i3}) P(J_i = 1)$. Equation (6.4) is obtained because $P(C_i = 1 | J_i = 1, F_{i1} \vee F_{i2} \vee F_{i3} = 1) = 1$. Equation (6.5) is based on the fact that:

$$P(J_i = 1 | C_{i-1} = 1) = \frac{P(J_i = 1, C_{i-1} = 1)}{P(C_{i-1} = 1)} = 1$$

$$P(J_i = 1 | C_{i-1} = 0) = \frac{P(J_i = 1, C_{i-1} = 0)}{P(C_{i-1} = 0)} = 0$$

The iterative procedure goes:

$$P(C_1 = 1) = \sum_{I_A(F_{11}, F_{12}, F_{13})} \frac{1}{2^3}$$

$$P(C_2 = 1) = \sum_{I_A(F_{21}, F_{22}, F_{23})} \frac{1}{2^3} P(C_1 = 1)$$

$$= \sum_{I_A(F_{21}, F_{22}, F_{23})} \sum_{I_A(F_{11}, F_{12}, F_{13})} \frac{1}{2^{3 \times 2}}$$

$$\vdots$$

$$P(C_n = 1) = \sum_{I_A(F_{n1}, F_{n2}, F_{n3})} \frac{1}{2^3} P(C_{n-1} = 1)$$

$$= \sum_{I_A(F_{n1}, F_{n2}, F_{n3})} \cdots \sum_{I_A(F_{11}, F_{12}, F_{13})} \frac{1}{2^{3n}}$$

Let $\sharp C_n$ be the number of true assignments that satisfy *1P3SAT(4)*, we have:

$$P(C_n = 1) = \frac{\sharp C_n}{2^{3n}}$$

Our proof is complete. □

Theorem 8. Given the approach of reduction from instantiation $S = (U, C^*), U = (X_1, ..., X_{3n}), C^* = (C_1^*, ..., C_n^*)$ to a *MBC* in Theorem 4(b), the $P(C_n = 1)$ satisfies:

$$P(C_n = 1) = \frac{\sharp C_n}{2^L}$$

Where $\sharp C_n$ is the number of true assignments that satisfy *1P3SATV(4)* and L is the number of chains in the feature graph.
Proof. Given the construction two, we show the structure of the generated *MBC* in Figure 7. The probability of $P(C_n = 1)$ satisfies:

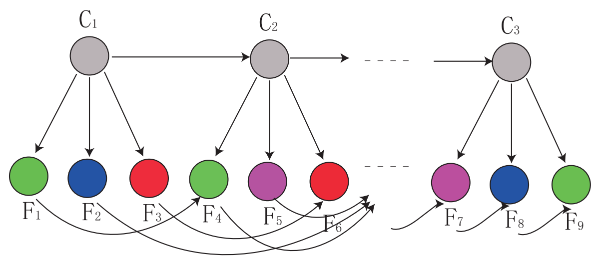


Figure 7: The schematic view of reconstructed *MBC* by construction 2.

$$P(C_i = 1) = \sum_{C_1 ... C_{i-1}, F_1 ... F_{3i}} P(C_i = 1, C_{i-1} ... C_1, F_{3i} ... F_1)$$

$$= \sum_{C_1 .. C_{i-1}, F_1 .. F_{3i}} P(C_i = 1 | C_{i-1} .. C_1, F_{3i} .. F_1) P(C_{i-1} ... C_1, F_{3i} ... F_1)$$

$$= \sum_{C_1 .. C_{i-1}, F_1 .. F_{3i}} P(C_i = 1 | C_{i-1}, F_{3i}, F_{3i-1}, F_{3i-2}) P(C_{i-1} ... C_1, F_{3i} ... F_1)$$

$$= \sum_{C_1 ... C_{i-2}, F_1 ... F_{3i}} P(C_i = 1 | C_{i-1} = 1, F_{3i}, F_{3i-1}, F_{3i-2}) P(C_{i-1} = 1, C_{i-2} ... C_1, F_{3i} ... F_1)$$

$$= \sum_{I_A(F_{3i}, F_{3i-1}, F_{3i-2})} [\sum_{C_1 ... C_{i-2}, F_1 ... F_{3i-3}} P(C_{i-1} = 1, C_{i-2} ... C_1, F_{3i} ... F_1)]$$

Thus:

$$P(C_i = 1) = \sum_{I_A(F_{3i},F_{3i-1},F_{3i-2})} \sum_{I_A(F_{3i-3},F_{3i-4},F_{3i-5})} \cdots \sum_{I_A(F_3,F_2,F_1)} P(F_{3n}, F_{3n-1}, F_{3n-2}, ..., F_1) \tag{7.6}$$

If $(F_1 = f_1, ..., F_{3n} = f_{3n})$ is a true assignment that satisfies the instance S in *1P3SAT(4)*, variables in the same chains of the feature graph G_F take the same value. We have:

$$\begin{aligned} & P(F_{3n}, F_{3n-1}, F_{3n-2}, ..., F_1) \\ & = \Pi_{i=1}^{L} P(X_i^1, X_i^2, ..., X_i^{L_i}) \quad (since\ X_i^j \in \{F_1, ..., F_{3n}\}\ and\ \textstyle\sum_{i=1}^{L} L_i = 3n) \\ & = \Pi_{i=1}^{L} P(X_i^1) \Pi_{j=2}^{L_i} P(X_i^j | X_i^{j-1}) \quad (since\ X_i^1 \to X_i^2 \to \cdots \to X_i^{L_i}) \\ & = \Pi_{i=1}^{L} P(X_i^1) \quad (since P(X_i^j = 1 | X_i^{j-1} = 1) = P(X_i^j = 0 | X_i^{j-1} = 0) = 1) \\ & = \frac{1}{2^L} \end{aligned} \tag{6.7}$$

Taking the equation (6.7) into (6.6) leads to:

$$P(C_i = 1) = \sum_{I_A(F_{3i},F_{3i-1},F_{3i-2})} [\cdots [\sum_{I_A(F_3,F_2,F_1)} \frac{1}{2^L}] \cdots]$$

Let $\sharp C_n$ be the number of true assignments that satisfy S in *1P3SATV(4)*, we have:

$$P(C_n = 1) = \frac{\sharp C_n}{2^L}. \qquad \square$$

References

[1] Concha B, Guangdi L, Pedro L, "Multi-dimensional classification with Bayesian networks" International Journal of Approximate Reasoning. 52(6): 705–727 (2011)

[2] Cooper GF, "Probabilistic inference using belief networks is NP-hard", Paper No. SMI-87-0195. Knowledge Systems Laboratory, Stanford University. Stanford, CA, USA

[3] P. Dagum and M. Luby, "Approximating probabilistic inference in Bayesian belief networks", IEEE Transactions on Pattern Analysis and Machine Intelligence,15: 246-255 (1993).

[4] M.L. Littman, J. Goldsmith, M. Mundhenk, "The computational complexity of probabilistic planning", J. Artificial Intelligence Res. 9: 1-36 (1998).

[5] Ashraf M. Abdelbar, Stephen T. Hedetniemi, Sandra Mitchell Hedetniemi, "The complexity of approximating MAPs for belief networks with bounded probabilities". Artif. Intell. 124(2): 283-288 (2000).

[6] James D.Park and Adnan Darwiche, "Approximating MAP using Local Search", In Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI), 403-410

[7] James D.Park, Adnan Darwiche. "Complexity Results and Approximation Strategies for MAP Explanations", Journal of Artificial Intelligence Research 21: 101-133 (2004).

[8] Solomon Eyal Shimony, Carmel Domshlak. "Complexity of probabilistic reasoning in directed-path singly-connected Bayes networks", Artif. Intell. 151(1-2): 213-225 (2003).

[9] Wu, Dan; Butz, C.J., "On the Complexity of Probabilistic Inference in Singly Connected Bayesian Networks", Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, 581-590, (2005).

[10] Vasanth Krishna Namasivayam, Animesh Pathak and Viktor K. Prasanna "Parallelizing Exact Inference in Bayesian Networks" 17Guan, Dah-Jyu. "Generalized Gray Codes with Applications". Proc. Natl. Sci. Counc. Repub. Of China (A) 22: 841-848 (1998)

[11] James D.Park and Adnan Darwiche, "Solving MAP Exactly using Systematic Search", In Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence(UAI), pages 459-468. (2002)

[12] Xiaoxun Sun,Marek J.Drudzel, Changhe Yuan. "Dynamic Weighting A* Search-based MAP Algorithm for Bayesian Network" IJCAI: 2385-2390 (2007).

[13] Van der Gaag,L,C,.de Waal, P,R: Multi-dimensional Bayesian network classifiers, In Studeny,M.Vomlel, J.(eds.) Proceeding of the Third European workshop in probabilistic Graphical Models. Prague,pp. 107-114 (2006).

[14] Peter R.de Waal and Linda C.van der Gaag, "Inference and learning in Multi-dimensional Bayesian Network Classifiers". Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference (2007).

[15] Rosa Blanco, Inaki Inza, Marisa Merino, Jorge Quiroga, Pedro Larranaga. "Feature selection in Bayesian classifiers fro the prognosis of survival of cirrhotic patients treated with TIPS", Journal of Biomedical Informatics 38:376-3 (2005).

[16] T. J. Schaefer. "The complexity of satisfiability problems". In Proceedings 10th Symposium on Theory of Computing, pages 216-226. ACM Press, (1978).

[17] M. Gamy, D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, New York, (1979).

[18] Christos H. Papadimitriou. "Computational Complexity". Addison Wesley,December 10, (1993).

[19] Ashraf M. Abdelbar, Sandra M. Hedetniemi, "Approximating MAPs for belief networks is NP-hard and other theorems", Artificial Intelligence, v.102 n.1, p.21-38, June (1998).

[20] U. Fischbacher, "Finding the maximum a posteriori probability (MAP) in a Bayesian taxonomic key is NP-hard", J. Math. Biol. 34:926-936, (1996).

[21] Guan, Dah-Jyu. "Generalized Gray Codes with Applications". Proc. Natl. Sci. Counc. Repub. Of China (A) 22: 841-848 (1998).